ELEC70037
Topics in Large Dimensional Data Processing
Exercise

Wei Dai

November 16, 2023

EEE Department, Imperial College London

Contents

Contents i

I. Teaching Organisaion 1

1. Teaching Organisation 2
1.1. Overview 2
1.2. Coursework 2
1.2.1. Schedule 2
1.2.2. Marking (for Coursework 1-3) 3
1.3. Software for Coursework 1-3 4
1.3.1. Julia for Programming 4

2. Feedback 5
2.1. You Said We Did 5
2.2. Your Feedback 5
2.3. Our Feedback to You 5

3. Important Notes 6
3.1. Plagiarism 6
3.2. Notes for Coursework Submission 6
3.2.1. Handling Solutions 6

II. Coursework 1 8

4. Linear Algebra: Basics 9
4.1. Trace and Inner Product of Matrices 9
4.2. The Matrix of the Linear Map 9
4.3. Adjoint Operator 10

5. Least Squares 14
5.1. The Netflix Problem 14
5.2. Estimation Accuracy 14
5.3. LS Recovery from Convolution 15
5.4. LS Recovery for MRI 16

III. Coursework 2 17

6. Proximal Operator 18

7. Sparse Inverse Problems 20
7.1. Greedy Algorithms 20
7.2. Proximal Gradient Method 20
7.3. MRI CS Recovery: Wavelet 20
7.4. MRI CS Recovery: DCT 21
7.5. MRI CS Recovery: Wavelet and DCT 21

IV. Coursework 3 **23**

8. Real Sparse Problems **24**
8.1. Proximal Operator (Continued) 24
8.2. MRI CS Recovery: ADMM 25
8.3. Blind Deconvolution: Convex Relaxation 27
8.4. Blind Deconvolution 28

V. Coursework 4 **30**

9. Learn and Tell Coursework Guideline **31**
9.1. Coursework Format 31
9.1.1. Presentation Sessions 31
9.2. Marking Scheme 32
9.2.1. Marking Criteria for Presentations 32
9.3. Topic Selection 32
9.3.1. Possible Topics 32

Part I.

Teaching Organisaion

1. Teaching Organisation

1.1. Overview

- Lectures
  - Every Monday between 9/10 and 27/11, 14:00-15:50, Room 305.
- Office hours
  - Every Monday between 9/10 and 27/11, 12:00-13:00, Room 503.
- Communications: Should you have any questions or inquiries:
  - Attend the designated office hours.
  - Post them on the Q&A channel.
  - For private matters, please contact us at `weidai.gta.course@gmail.com`.
  - Refrain from using my college email address, as it might result in your email getting overlooked.
- Assessment
  4 coursework: each counts 25%
  - Coursework 1-3 are composed of technical questions.
  - Coursework 4 is a "Learn and tell" presentation.

1.2. Coursework

1.2.1. Schedule

Table 1.1.: Coursework Schedule

	Tasks	Due Date/Time
Coursework 1	Assignment	(Week 3) 16/10
	Due	(Week 5) 30/10 09:59
	Feedback	(Week 6) 6/11
Coursework 2	Assignment	(Week 5) 30/10
	Due	(Week 7) 13/11 09:59
	Feedback	(Week 8) 20/11
Coursework 3	Assignment	(Week 8) 20/11
	Due	(Week 11) 11/12 09:59
	Feedback	Marks in Jan. 2024
Coursework 4	Assignment	(Week 5) 30/10
	Group/Topic decisions	(Week 8) 20/11
	Slides Due	(Week 10) 4/12 08:59
	Presentations	(Week 10) 4-6/12
	Peer marking	(Week 10) 7/12 23:59
	Feedback	(Week 11) 11/12

Note that coursework 4 will be finished before coursework 3 ends.

The procedure for coursework 1-3 is given below.

- Each individual student will receive their own data file from the GTAs.
- Students can form groups in finishing the coursework.
- Before the due time, students need to submit their individual coursework submissions, and specify their group and their contributions to the group. Please note that the relevant files will not be allowed to change after the deadline.
- GTAs will mark individual coursework submission. Let $M_{i,0}$ denote the raw mark received by the student i. Their adjusted mark will be calculated based on the equation (1.1). The adjusted mark will be returned to the student.
- GTAs will choose some coursework questions to discuss. The students who are chosen to lead specific discussions will be given extra marks in order to encourage participation.

The process of coursework 4 is detailed in Chapter 9.

1.2.2. Marking (for Coursework 1-3)

The marking formula for the first three coursework is given by

$$M_i = \min\left(\left(\sum_{k\in\mathcal{G}_l} M_{k,0}\right) * C_i * W_l, M_{i,0}\right), \tag{1.1}$$

where

- M_i is the adjusted mark received by the student i,
- $M_{k,0}$ is the raw mark obtained by the student k,
- $\sum_{k\in\mathcal{G}_l} M_{k,0}$ is the total raw mark obtained by the group $\mathcal{G}_l$,
- C_i is the contribution of the student i in the group,[1]
- W_l is the weighting coefficient for the group $\mathcal{G}_l$, depending on the size of the group,
- and we apply the min function to ensure the adjusted mark does not exceeds the raw mark.

1: Note that

$$\sum_{k\in\mathcal{G}_l} C_k = 1.00 = 100\%.$$

The weighting coefficients for groups are detailed in Table 1.2. There are several considerations behind the design.

- We encourage both individual work and team work.
- The default size of the group is 3 or 4.
- Students from the same group may share the same codes.
- **Cheating offences and plagiarism** are taken very seriously and are dealt with according to the College's Academic Misconduct Policy and Procedure.

Size of the group	1	2	3	4	5	≥ 6
Weighting coefficient W	1.00	0.96	0.92	0.90	0.70	0.50

Table 1.2.: The weighting coefficient W for the group is decided by the size of the group.

Remark 1.2.1 (Extra marks) If

1. a student indicates that they are comfortable to share their

coursework solutions in the feedback/discussion sessions,
2. their coursework solutions are indeed chosen by GTAs and the lecturer for feedback/discussion, and
3. the student participates in explaining the key ideas behind their solutions,

then extra marks will be given the student.
The amount of extra marks given to the student will be decided by the GTAs and the lecturer, based on the quality of the solutions and the explanation of key ideas. The amount of extra marks will be announced to the student.

1.3. Software for Coursework 1-3

The coursework requires Julia programming. The coursework submission files (except coursework 4) include a Jupyter notebook file and a data file in JLD2 format. We recommend VSCode as the default editor.

The software that you need to install/have

- Jupyter Notebook
  Download and install the package management software Anaconda. By default, it will install Python and Jupyter Notebook to your system.
- Julia programming language
  - Download and install the current stable version of Julia.
  - Check whether Julia has been installed properly: Run Julia interactive session (a.k.a. REPL) by following the instructions.
- VSCode
  - Download and install VSCode.
  - We need to install some extensions for VSCode. See here for how to install VSCode extensions.
    - Anaconda should have automatically installed VSCode Extensions `Python` and `Pylance` for you. If not, install them.
    - Install extension `Julia` for Julia programming in VSCode. See here and here for more details.

1.3.1. Julia for Programming

- A collection of tutorials.
- Introduction to Scientific Programming and Machine Learning with Julia.
- A tutorial in pdf format.

Feedback | 2.

2.1. You Said We Did

Based on students' feedback in the past several years, we have done the following changes

- Lecture notes
  - We have changed the lecture notes from slides to a book format.
    - Contents are more self-contained.
    - We use `kaobook` LaTeX template so that we have enough space to make notes in the lectures.
  - Less theory and more examples
- Assessment
  - The format has been changed from a final exam to four coursework.
    - Emphasise a lot more on the applications of theory.
    - Help students horn analysis and programming skills.
  - Significantly reduce the weighting of peer marking.
  - Introduce and refine the coursework contents and the marking scheme to encourage both individual efforts and teamworking
    - Each individual student get their own data.
    - Each individual student can specify their own contributions to the group.
    - Individual student's mark depends on the size of the group.

2.2. Your Feedback

Towards the end of the term, please do the **MEQ** (previously known as SOLE) questionaries from the college.

Towards the end of the term, GTAs may contact you for our own questionaries for this module.

2.3. Our Feedback to You

- We target at finishing the coursework marking within 10 working days. We will return the relevant information to you.
- In the lectures, we will discuss some interesting/challenging coursework questions together.

3. Important Notes

3.1. Plagiarism

PLAGIARISM/COLLUSION DECLARATION

Coursework submitted for assessment must be the original work of you and your group. Assignments are subjected to regular checks for plagiarism and/or collusion. Plagiarism is the presentation of another person's thoughts or words (those outside your group) as if they were your own. Collusion involves obtaining help from someone outside your group to complete your work. In preparing your coursework, you should not seek help, or copy from any other person or source, including the Internet, without proper and explicit acknowledgement.

There is a procedure in place for you to declare individual contributions within your group for courswork. You must declare the contributions fairly and accurately.

You must not disclose your solutions or insights related to coursework with anyone else, including future students or the Internet.

By acknowledging the the statements above, you are declaring that both this and all subsequent pieces of coursework are, and will remain, the original work of you and your group.

- Submissions will not be accepted without the aforementioned declaration.
- Members of a group are deemed to have collective responsibility for the integrity for work submitted and are liable for any penalty imposed, proportionate to their contributions.

3.2. Notes for Coursework Submission

- Each registered student will get a data file. The data in the data file can be unique.
- Each registered student needs to submit the solutions related to their own data, no matter whether they are in groups or not.

3.2.1. Handling Solutions

In our coursework, we use the following convention.

- If the solution to a question is a unique integer, you need to assign an integer value to your solution variable.
- If the solution to a question does not exist, you need to create a 1-D array of length zero.
- If the solution to a question is not unique, you need to create a 1-D array, of which the length is the number of distinct solutions, and then specify the values in the **ascending** order.

Examples:

```
x = 3;
y = Array{Int64,1}(undef,0);
z = Array{Int64,1}(undef,3);
z[1] = 3; z[2] = 4; z[3] = 5;
```

Part II.

Coursework 1

Linear Algebra: Basics 4.

4.1. Trace and Inner Product of Matrices

Give a sqaure matrix $A \in \mathbb{R}^{n\times n}$, its trace is defined as the sum of the diagonal elements:

$$\operatorname{trace}(A) := \sum_{i=1}^{n} A_{i,i}.$$

Use the parameters in your data file to answer the following questions.

1. Let $A, B \in \mathbb{R}^{m\times n}$.
   a) Identify the dimension of $AB^\top$. [2]
   b) Identify the dimension of $A^\top B$. [2]
   c) Is the following statement true or false?
   $$\operatorname{trace}\left(AB^\top\right) = \operatorname{trace}\left(A^\top B\right).$$
   [1]
2. For any two matrices $A, B \in \mathbb{R}^{m\times n}$, define the **inner product** between them:
   $$\langle A, B\rangle := \operatorname{trace}\left(A^\top B\right).$$
   Is the following statement true or false?
   $$\langle A, B\rangle = \langle \operatorname{vect}(A), \operatorname{vect}(B)\rangle.$$
   [1]
3. Let $A \in \mathbb{R}^{k\times l}$ and $B \in \mathbb{R}^{m\times n}$.
   a) Specify the dimensions of U and V such that
   $$\left\langle UAV^\top, B\right\rangle$$
   is well-defined. [4]
   b) Is the following statement true or false?
   $$\left\langle UAV^\top, B\right\rangle = \left\langle A, U^\top BV\right\rangle.$$
   [1]

4.2. The Matrix of the Linear Map

Remark 4.2.1 A linear map (or linear transformation) between two finite-dimensional vector spaces can always be represented by a matrix, called the matrix of the linear map.

Many of the coursework questions are using the above fact.

In the following problem, the linear map is from a vector space of matrices of dimension $m \times n$ to a vector space of matrices of dimension $s \times t$. This linear map can be represented by a matrix of proper dimension.

Definition 4.2.1 *Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{p \times q}$. The Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B}$ is the $pm \times qn$ (block) matrix given by*

$$\boldsymbol{A} \otimes \boldsymbol{B} := \begin{bmatrix} A_{1,1}\boldsymbol{B} & \cdots & A_{1,n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ A_{m,1}\boldsymbol{B} & \cdots & A_{m,n}\boldsymbol{B} \end{bmatrix}.$$

1. Let

$$\begin{aligned} \mathscr{C} : \mathbb{R}^{m \times n} &\longrightarrow \mathbb{R}^{s \times t} \\ \boldsymbol{X} &\mapsto \boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}\boldsymbol{B}. \end{aligned}$$

Find the matrix of the linear map, denoted by $\boldsymbol{C}$, using *Kronecker product* such that

$$\text{vect}(\boldsymbol{Y}) = \boldsymbol{C}\text{vect}(\boldsymbol{X}).$$

1 [2]

1: $\boldsymbol{C} = \boldsymbol{B}^{\mathsf{T}} \otimes \boldsymbol{A}$

Remark 4.2.2 It is important to note that although the above linear map can be represented by the matrix $\boldsymbol{C}$, the evaluation of the linear map in practice is via $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}\boldsymbol{B}$ rather than `\reshape( C * X[:], s, t )`.

The approach to evaluate the linear map without using its matrix representation is sometimes called the **matrix-free** approach.

In the following coursework questions, sometimes you are asked to find the matrix of the linear map, sometimes you are expected to evaluate the linear map without using the matrix representation.

4.3. Adjoint Operator

Let $\mathscr{A}$ be a linear operator mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$. It always can be represented by a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, that is,

$$\begin{aligned} \mathscr{A} : \mathbb{R}^{n} &\longrightarrow \mathbb{R}^{m} \\ \boldsymbol{x} &\mapsto \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \end{aligned}$$

Definition 4.3.1 *The adjoint of the linear operator $\mathscr{A}$, denoted by $\mathscr{A}^*$, is defined via*

$$\langle \mathscr{A}(\boldsymbol{x}), \boldsymbol{y} \rangle = \langle \boldsymbol{x}, \mathscr{A}^*(\boldsymbol{y}) \rangle$$

for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$.

It can be verified that

$$\mathscr{A}^*(\boldsymbol{y}) = \boldsymbol{A}^{\mathsf{T}}\boldsymbol{y}.$$

If $\mathcal{A} : \mathbb{C}^n \to \mathbb{C}^m$, then $\mathcal{A}$ is generally represented by a matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$. That is,

$$\mathcal{A}^*(\boldsymbol{y}) = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{y},$$

where the superscript H denotes 'conjugate transpose'.

The following exercise questions involve identification of linear operators and their adjoint operators. One way to check the correctness of your adjoint operator is to see whether the equality

$$\langle \mathcal{A}(\boldsymbol{x}), \mathcal{A}(\boldsymbol{y}) \rangle = \langle \boldsymbol{x}, \mathcal{A}^* (\mathcal{A}(\boldsymbol{y})) \rangle$$

holds.

1. Consider the subsampling operator $\mathcal{P}_\Omega$ where $\Omega = \{i_1, i_2, \cdots, i_m\}$ is a subset of $[1 : n] := \{1, 2, \cdots, n\}$ $(m \le n)$, defined as

$$\begin{aligned} \mathcal{P}_\Omega :\ & \mathbb{C}^n \to \mathbb{C}^m \\ & \boldsymbol{x} \mapsto \boldsymbol{y} \text{ with } y_i = x_{i_l}. \end{aligned}$$

   a) Write a function to find the corresponding matrix representations of $\mathcal{P}_\Omega$ and $\mathcal{P}_\Omega^*$.
   Use the data in the data file for a test. [4]
   b) Write a function to evaluate $\boldsymbol{y} = \mathcal{P}_\Omega(\boldsymbol{x})$ without using the matrix representation of the linear map.
   Use the data in the data file for a test. [2]
   c) Write a function to evaluate $\boldsymbol{z} = \mathcal{P}_\Omega^*(\boldsymbol{y})$ without using the matrix representation of the linear map.
   Use the data in the data file for a test. [2]

2. Consider the Hankel operator given by

$$\mathcal{H}_{p,q} :\ \mathbb{C}^{p+q-1} \to \mathbb{C}^{p\times q}$$
$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{p+q-1} \end{bmatrix} \mapsto \boldsymbol{H} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_q \\ x_2 & x_3 & x_4 & \cdots & x_{q+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_p & x_{p+1} & x_{p+2} & \cdots & x_{p+q-1} \end{bmatrix}.$$

   Or equivalently,

$$H_{m,n} = x_{m+n-1}.$$

   a) Write a function to find the corresponding matrix representations of $\mathcal{H}_{p,q}$ and $\mathcal{H}_{p,q}^*$.
   Use the data in the data file for a test. [4]
   b) Write a function to evaluate $\boldsymbol{H} = \mathcal{H}_{p,q}(\boldsymbol{x})$ without using the matrix representation of the linear map.
   Use the data in the data file for a test. [2]
   c) Write a function to evaluate $\boldsymbol{z} = \mathcal{H}_{p,q}^*(\boldsymbol{H})$ without using the matrix representation of the linear map.
   Use the data in the data file for a test. [2]
   d) Evaluate $\boldsymbol{z}./\boldsymbol{x}$ where the symbol ./ denotes element division. [1]

3. Let $\boldsymbol{h} \in \mathbb{R}^m$ be given. For any $\boldsymbol{x} \in \mathbb{R}^n$, the corresponding convolu-

tion map is given by

$$\begin{aligned} \mathscr{C}_{\boldsymbol{h}} : \mathbb{R}^n &\to \mathbb{R}^{n+m-1} \\ \boldsymbol{x} &\mapsto \boldsymbol{y} = \boldsymbol{x} \circledast \boldsymbol{h} \\ & y[t] := \sum_{1 \le \tau \le t} x[t+1-\tau] \times h[\tau]. \end{aligned}$$

a) Write a function to find the corresponding matrix representations of $\mathscr{C}_{\boldsymbol{h}}$ and $\mathscr{C}_{\boldsymbol{h}}^*$.
Use the data in the data file for a test. [4]

b) Write a function to evaluate $\boldsymbol{y} = \boldsymbol{x} \circledast \boldsymbol{h} = \mathscr{C}_{\boldsymbol{h}}(\boldsymbol{x})$ without using the matrix representation of the linear map.
Use the data in the data file for a test. [2]

c) Write a function to evaluate $\boldsymbol{z} = \mathscr{C}_{\boldsymbol{h}}^*(\boldsymbol{y})$ without using the matrix representation of the linear map.
Use the data in the data file for a test. [2]

4. Let $\boldsymbol{h}$ and $\boldsymbol{x}$ be sequences of infinite length that are periodic with a period n, i.e.,

$$\begin{aligned} h[k] &= h[l], \quad \forall k, l \in \mathbb{Z} \text{ with } k - l \equiv 0 \pmod{n}, \\ x[k] &= x[l], \quad \forall k, l \in \mathbb{Z} \text{ with } k - l \equiv 0 \pmod{n}. \end{aligned}$$

With slight abuse of notations, use $\boldsymbol{h} \in \mathbb{R}^n$ and $\boldsymbol{x} \in \mathbb{R}^n$ to represent these two infinite sequences.
Define the cyclic convolution of them as

$$\begin{aligned} \mathscr{C}_{\boldsymbol{h},n} : \mathbb{R}^n &\to \mathbb{R}^n \\ \boldsymbol{x} &\mapsto \boldsymbol{y} = \boldsymbol{x} \circledast_n \boldsymbol{h} \\ & y[t] := \sum_{1 \le \tau \le n} x[t+1-\tau] \times h[\tau]. \end{aligned}$$

a) Write a function to find the corresponding matrix representations of $\mathscr{C}_{\boldsymbol{h},n}$ and $\mathscr{C}_{\boldsymbol{h},n}^*$.
Use the data given in the data file as a test. [4]

b) Write a function to evaluate $\boldsymbol{y} = \boldsymbol{x} \circledast_n \boldsymbol{h} = \mathscr{C}_{\boldsymbol{h},n}(\boldsymbol{x})$ without using the matrix representation of the linear map. Use the Julia function `DSP.conv`.
Use the data given in the data file as a test. [2]

c) Write a function to evaluate $\boldsymbol{z} = \mathscr{C}_{\boldsymbol{h},n}^*(\boldsymbol{y})$ without using the matrix representation of the linear map. Use the Julia function `DSP.conv`.
Use the data given in the data file as a test. [2]

It is important to note that `DSP.conv` is implemented using the *convolution theorem*.
The seminal *convolution theorem* states that the Fourier transform of a convolution of two signals is the pointwise product of their Fourier transforms. That is,

$$\boldsymbol{y} = \boldsymbol{x} \circledast_n \boldsymbol{h} \quad \Leftrightarrow \quad \mathscr{F}(\boldsymbol{y}) = \mathscr{F}(\boldsymbol{x}) \odot \mathscr{F}(\boldsymbol{h}).$$

For large dimensional problems, you can feel the speed difference between your own convolution and the Julia `DSP.conv` implementations.

5. (A linear map related to Magnetic Resonance Imaging (MRI) sub-

sampling)
This question requires background in

- 2D Discrete Fourier Transform. See Wikipedia page `https://en.wikipedia.org/wiki/Discrete_Fourier_transform`, Sections 'Definition', 'Inverse Transform', and 'Multidimensional DFT'. Also see `https://www.matecdev.com/posts/julia-fft.html#fftshift-and-fftfreq`.
- `fftshift`: See `https://www.matecdev.com/posts/julia-fft.html#fftshift-and-fftfreq`.

Let $X \in \mathbb{R}^{m \times n}$ be an MRI image. The MRI data compressed sensing can be mathematically expressed as

$$y = \mathcal{P}_\Omega \circ \mathcal{S} \circ \mathcal{F}_{2D}(X),$$

where

- $\mathcal{F}_{2D}$ denotes the 2D Discrete Fourier Transform,
- $\mathcal{S}$ denotes the 2D `fftshift` operator (which is the straightforward extension of 1D `fftshift`), and
- $\mathcal{P}_\Omega$ is the subsampling operator.

In the following, you are allowed to use `FFTW.fft`, `FFTW.ifft`, and `FFTW.fftshift`.
Make sure that your function works for both odd and even m and n.

a) Write a function to evaluate

$$y = \mathcal{P}_\Omega \circ \mathcal{S} \circ \mathcal{F}_{2D}(X)$$

without using matrix representation of the linear map.
The outputs of your function must include $O_1 = \mathcal{F}_{2D}(X)$, $O_2 = \mathcal{S} \circ \mathcal{F}_{2D}(X)$, and $O_3 = \mathcal{P}_\Omega \circ \mathcal{S} \circ \mathcal{F}_{2D}(X)$.
Use the data in the data file for a test. [6]

b) Write a function to evaluate

$$Z = (\mathcal{P}_\Omega \circ \mathcal{S} \circ \mathcal{F}_{2D})^* (y)$$

without using matrix representation of the linear map.
The outputs of your function must include $O_1 = \mathcal{P}_\Omega^*(y)$, $O_2 = \mathcal{S}^* \circ \mathcal{P}_\Omega^*(y)$, and $O_3 = \mathcal{F}_{2D}^* \circ \mathcal{S}^* \circ \mathcal{P}_\Omega^*(y)$.
Use the data in the data file for a test. [6]

5. Least Squares

5.1. The Netflix Problem

1. The task is to recover an unknown matrix $\boldsymbol{X}$ from its partial observations given by

$$\boldsymbol{y} = \mathcal{P}_\Omega(\boldsymbol{X}) + \boldsymbol{w},$$

where $\boldsymbol{w}$ denotes noise.
The alternating minimization method to solve the Netflix problem is stated below. Suppose the prior information that the rank of $\boldsymbol{X}$ is at most r. Then one would find the matrices $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times r}$ such that $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\top$. In the k-th iteration of the alternating minimization, one solves the following two least squares sequentially:

$$\boldsymbol{V}^{k+1} = \arg\min_{\boldsymbol{V}} \left\| \boldsymbol{y} - \mathcal{P}_\Omega\left(\boldsymbol{U}^k\boldsymbol{V}^\top\right)\right\|_2^2,$$
$$\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}} \left\| \boldsymbol{y} - \mathcal{P}_\Omega\left(\boldsymbol{U}\boldsymbol{V}^{k+1,\top}\right)\right\|_2^2.$$

The partial observations $\boldsymbol{y}$ and the index sets of the observed entries Ω are given in the data file. Suppose that $\operatorname{rank}(\boldsymbol{X}) = 10$.

a) Write a function to update $\boldsymbol{V}$ for a given $\boldsymbol{U}$. Use the $\boldsymbol{U}_0$ in the data file for a test. [2]
b) Write a function to update $\boldsymbol{U}$ for a given $\boldsymbol{V}$. Use the $\boldsymbol{V}$ from the previous sub-question for a test. [2]
c) Write a function for the alternating miminimization process. Use the $\boldsymbol{U}_0$ in the data file as the initial point. Run a test. [2]

5.2. Estimation Accuracy

1. Alice, Bob, and Charlie's jobs are to estimate unknown $\boldsymbol{x}$ from their respective measurement vector $\boldsymbol{y}$ given by

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{w},$$

by solving the minimization problem

$$\hat{\boldsymbol{x}} = \arg\ \min_{\boldsymbol{z}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|^2.$$

Daniel knows the ground truth $\boldsymbol{x}_0$ and has the power to choose the noise vector $\boldsymbol{w}$ with $\|\boldsymbol{w}\|_2 = 1$ in generating $\boldsymbol{y}$.

a) Daniel is a friend of Alice. He would like to choose a noise vector $\boldsymbol{w}$ so that the estimation error

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|^2$$

is minimised.
Find the best $\boldsymbol{w}$ and compute the corresponding estimation error for Alice. [2]

b) Daniel is a foe of Bob. He would like to choose a noise vector $\boldsymbol{w}$ so that the estimation error

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|^2$$

is maximised.
Find the best $\boldsymbol{w}$ and compute the corresponding estimation error for Bob. [2]

c) For Charlie, Daniel randomly generates a noise vector $\boldsymbol{w}$ with i.i.d. Gaussian entries and then normalises it so that $\|\boldsymbol{w}\|_2 = 1$ (i.e., $\boldsymbol{w}$ has unit l_2 norm and statistically invariant under rotations). Find the expectation of the estimation error, i.e.,

$$\mathrm{E}\left[\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|^2\right]$$

that Charlie may expect. [2]

5.3. LS Recovery from Convolution

Let

$$\boldsymbol{y} = \mathscr{P}_\Omega \left(\boldsymbol{h} \circledast_n \boldsymbol{x}\right) + \boldsymbol{w},$$

where $\boldsymbol{y}$, $\boldsymbol{h}$, and Ω are given. The task is to recover $\boldsymbol{x}$ from the noisy and partial observation vector $\boldsymbol{y}$.

Towards this goal, we solve the following least squares problem

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \ \frac{1}{2} \|\boldsymbol{y} - \mathscr{P}_\Omega \left(\boldsymbol{h} \circledast_n \boldsymbol{x}\right)\|_2^2 + \frac{\alpha}{2} \|\boldsymbol{x}\|_2^2,$$

where $\alpha > 0$ is a small positive number to ensure the matrix involved in the least square problem is *not* rank deficient.

The standard form of a least squares problem is given by

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \ \frac{1}{2} \boldsymbol{x}^\mathsf{T} \boldsymbol{A} \boldsymbol{x} + \boldsymbol{b}^\mathsf{T} \boldsymbol{x} + c.$$

1. (Matrix based approach)

   a) Write a Julia function to compute the corresponding $\boldsymbol{A}$ and $\boldsymbol{b}$ from the data given in the data file. [4]

   b) Use Julia's operator \to get $\hat{x}$. [1]

2. (Matrix free approach)

   a) Write a function that uses `DSP.conv` to compute $\boldsymbol{u}^\mathsf{T} \boldsymbol{A} \boldsymbol{v}$ and $\boldsymbol{b}^\mathsf{T} \boldsymbol{u}$ for arbitrary $\boldsymbol{u}$ and $\boldsymbol{v}$ of appropriate dimension. Note that $\boldsymbol{A}$ and $\boldsymbol{b}$ are defined as the previous question. Your implementation should not use the explicit forms of $\boldsymbol{A}$ and $\boldsymbol{b}$. Use the data in the data file for a test. [4]

b) Use the Julia library `LinearOperators.jl` so as to use matrix-free functions to support matrix operations.
Use `IterativeSolvers.cg` and your matrix free linear operator to solve the aforementioned least squares problem.
Note the difference in runtime between the matrix approach and the matrix free approach. [4]

5.4. LS Recovery for MRI

Consider the dimension of the MRI image data, we use matrix free approach for least squares recovery of undersampled MRI data. The least squares problem under consideration is given by

$$\hat{\boldsymbol{X}} = \arg\min_{\boldsymbol{X}} \frac{1}{2} \|\boldsymbol{y} - \mathcal{P}_\Omega \circ \mathcal{S} \circ \mathcal{F}_{2D}(\boldsymbol{X})\|_2^2 + \frac{\alpha}{2} \|\boldsymbol{X}\|_F^2 .$$

With vectorization of $\boldsymbol{X}$, the above formulation can be written as the standard form of least squares:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x} + c.$$

1. Use the data given in the data file, find the observation vector $\boldsymbol{y}$. [2]
2. Write a Julia function that uses `FFTW.fft` and `FFTW.fftshift` to compute $\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{v}$ and $\boldsymbol{b}^\top \boldsymbol{z}$ for arbitrary and appropriate $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{z}$.
   Note the size of $\boldsymbol{X}$. Your implementation should use the matrix free approach. [6]
3. Use the Julia library `LinearOperators.jl` so as to use matrix-free functions to support matrix operations.
   Use `IterativeSolvers.cg` and your matrix free linear operator to solve the aforementioned least squares problem.
   [4]
   Note the difference between the groundtruth $\boldsymbol{X}$ and your least squares estimate $\hat{\boldsymbol{X}}$ due to the subsampling and the regularization term $\frac{\alpha}{2} \|\boldsymbol{X}\|_F^2$. In next courseworks, sparse recovery techniques will be applied to improve the recovery quality.

Part III.

Coursework 2

Proximal Operator 6.

1. Use the data in the data file to compute the solutions of the following optimization problems.

   a)

   $$\min_{x} \frac{1}{2} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x} + \frac{1}{2\gamma} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 .$$

   [2]

   b)

   $$\min_{x} \delta(\boldsymbol{a} \le \boldsymbol{x} \le \boldsymbol{b}) + \frac{1}{2\gamma} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 ,$$

   where $\boldsymbol{a} \le \boldsymbol{x} \le \boldsymbol{b}$ denotes element-wise inequalities.

   [2]

   c)

   $$\min_{x} \delta(\|\boldsymbol{x}\|_0 \le k) + \frac{1}{2\gamma} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 .$$

   [2]

   d)

   $$\min_{x} \|\boldsymbol{x}\|_0 + \frac{1}{2\gamma} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 .$$

   [2]

   e)

   $$\min_{X} \delta(\operatorname{rank}(\boldsymbol{X}) \le r) + \frac{1}{2\gamma} \|\boldsymbol{X} - \boldsymbol{Z}\|_F^2 .$$

   [2]

   f)

   $$\min_{X} \operatorname{rank}(\boldsymbol{X}) + \frac{1}{2\gamma} \|\boldsymbol{X} - \boldsymbol{Z}\|_F^2 .$$

   [2]

   g)

   $$\min_{X} \delta(\boldsymbol{X} \succeq 0) + \frac{1}{2\gamma} \|\boldsymbol{X} - \boldsymbol{Z}\|_F^2 ,$$

   where $\boldsymbol{X} \succeq 0$ denotes that $\boldsymbol{X}$ is a positive semi-definite matrix, i.e., all the eigenvalues of the symmetric matrix $\boldsymbol{X}$ are non-negative.

   [2]

   h)

   $$\min_{x} \delta(\|\boldsymbol{x}\|_2 = 1) + \frac{1}{2\gamma} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 .$$

   [2]

2. (Proximal operators related to ℓ_1-norm) Use the data in the data file to compute the solutions of the following optimization problems.

   a)

   $$\min_{\boldsymbol{x}} \lambda \|\boldsymbol{x}\|_1 + \frac{1}{2\gamma} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 .$$

   [2]

   b)

   $$\min_{\boldsymbol{x}} \lambda \|\boldsymbol{x}\|_1 + \frac{1}{2\gamma} \|\mathrm{diag}\,(\boldsymbol{a})\, \boldsymbol{x} - \boldsymbol{z}\|_2^2 ,$$

   where $\mathrm{diag}(\boldsymbol{a})$ turns the vector $\boldsymbol{a}$ into a diagonal matrix with the diagonal vector $\boldsymbol{a}$.

   [2]

   c)

   $$\min_{\boldsymbol{x}} \lambda \|\boldsymbol{x}\|_1 + \frac{1}{2\gamma} \|\boldsymbol{U}\boldsymbol{x} - \boldsymbol{z}\|_2^2 ,$$

   where the matrix $\boldsymbol{U}$ is a unitary matrix, i.e., $\boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}$.

   [2]

   d)

   $$\min_{\boldsymbol{x}} \lambda \|\boldsymbol{U}\boldsymbol{x}\|_1 + \frac{1}{2\gamma} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 ,$$

   where the matrix $\boldsymbol{U}$ is a unitary matrix, i.e., $\boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}$.

   [2]

7. Sparse Inverse Problems

7.1. Greedy Algorithms

1. Implement the OMP algorithm. Test its performance using the data given in the data file. [4]
2. Implement the SP algorithm. Test its performance using the data given in the data file. [4]
3. Implement the HIT algorithm. Test its performance using the data given in the data file. [4]

7.2. Proximal Gradient Method

Alice ang Bob are trying to solve the following optimization problem:

$$\min_{\boldsymbol{x}} \ \lambda \left\| \boldsymbol{x} \right\|_1 + \frac{1}{2\gamma} \left\| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{z} \right\|_2^2 ,$$

where the matrix $\boldsymbol{A}$ is invertible and its inverse is denoted by $\boldsymbol{A}^{-1}$.

1. Bob claims that the optimal solution admit the following closed form

   $$\hat{x}_i = \operatorname{sign}\left(\left(\boldsymbol{A}^{-1}\boldsymbol{z} \right)_i \right) \cdot \max\left(\left(\boldsymbol{A}^{-1}\boldsymbol{z} \right)_i , \lambda\gamma \right) .$$

   Write a function to implement Bob's closed form solution $\hat{\boldsymbol{x}}$ using the data given in the data file. [2]
   Compute the subgradient at the $\hat{\boldsymbol{x}}$ and check whether $\boldsymbol{0} \in \partial f(\hat{\boldsymbol{x}})$. [2]
2. Alice decides to solve this problem differently. She applies the proximal gradient method and uses Bob's solution as the initial point of her algorithm.
   Based on the data given in the data file, find the upper bound of the step size in the proximal gradient method. [2]
   Set the step size of Alice's proximal gradient method as half of its upper bound. Implement it. Run it for 5 iterations. Record the value of the objective function at the end of each iteration. [3]

7.3. MRI CS Recovery: Wavelet

Let $\boldsymbol{X}$ be an unknown MRI image. We consider the following MRI compressed sensing recovery formulation:

$$\begin{aligned} \hat{\boldsymbol{X}} &= \arg\min_{\boldsymbol{X}} \ f(\boldsymbol{X}) \\ &:= \arg\min_{\boldsymbol{X}} \ \lambda \left\| \mathscr{W}(\boldsymbol{X}) \right\|_1 + \frac{1}{2} \left\| \boldsymbol{y} - \mathscr{P}_\Omega \circ \mathcal{S} \circ \mathscr{F}_{2\mathrm{D}}(\boldsymbol{X}) \right\|_2^2 \\ &= \arg\min_{\boldsymbol{X}} \ \lambda \left\| \mathscr{W}(\boldsymbol{X}) \right\|_1 + \frac{1}{2} \left\| \boldsymbol{y} - \mathscr{A}(\boldsymbol{X}) \right\|_2^2 \end{aligned}$$

where $\|\cdot\|_1$ is the sum of the absolute values of the entries, $\mathscr{W}$ denotes the 2D discrete Daubechies wavelet transform (which can be represented by an orthonormal matrix) (See Wavelets.jl for Julia functions `dwt` and `idwt`), the operators $\mathscr{P}_\Omega, \mathcal{S}, \mathscr{F}_{2\mathrm{D}}$ have been defined in coursework 1, and for simplicity we have used the notation

$$\mathscr{A} = \mathscr{P}_\Omega \circ \mathcal{S} \circ \mathscr{F}_{2\mathrm{D}}.$$

We use proximal gradient method to solve the above optimization problem.

1. Use the $\boldsymbol{X}^0$ given in the data file to compute the gradient
$$\nabla_{\boldsymbol{X}} \left(\frac{1}{2} \|\boldsymbol{y} - \mathscr{A}(\boldsymbol{X}_0)\|_2^2 \right).$$
[3]
2. Let $\boldsymbol{X} \in \mathbb{R}^{n \times n}$. Set the step size $\tau = \frac{1}{2n}$. Find the $\boldsymbol{X}^1$ after the first iteration of the proximal gradient method. [3]
3. Implement the proximal gradient method. Run it until it converges. Draw your recovery. [4]

7.4. MRI CS Recovery: DCT

Let $\boldsymbol{X}$ be an unknown MRI image. We consider the following MRI compressed sensing recovery formulation:

$$\hat{\boldsymbol{X}} = \arg\min_{\boldsymbol{X}} \ \lambda \|\mathscr{D}(\boldsymbol{X})\|_1 + \frac{1}{2} \|\boldsymbol{y} - \mathscr{A}(\boldsymbol{X})\|_2^2$$

where $\mathscr{D}$ denotes the 2D discrete cosine transform (which can be represented by an orthonormal matrix) (See FFTW.jl for Julia functions `dct` and `idct`), and $\mathscr{A}$ is defined as in the previous question.

We use proximal gradient method to solve the above optimization problem.

1. Let $\boldsymbol{X} \in \mathbb{R}^{n \times n}$. Set the step size $\tau = \frac{1}{2n}$. Find the $\boldsymbol{X}^1$ after the first iteration of the proximal gradient method. [2]
2. Implement the proximal gradient method. Run it until it converges. Draw your recovery. [3]

7.5. MRI CS Recovery: Wavelet and DCT

Let $\boldsymbol{X}$ be an unknown MRI image. We consider the following MRI compressed sensing recovery formulation:

$$\begin{aligned}\hat{\boldsymbol{X}} = \arg\min_{\boldsymbol{X},\boldsymbol{Z}_1,\boldsymbol{Z}_2} \ & \lambda \|\boldsymbol{Z}_1\|_1 + \lambda \|\boldsymbol{Z}_2\|_1 \\ & + \frac{\alpha}{2} \|\boldsymbol{Z}_1 - \mathscr{W}(\boldsymbol{X})\|_2^2 + \frac{\alpha}{2} \|\boldsymbol{Z}_2 - \mathscr{D}(\boldsymbol{X})\|_2^2 \\ & + \frac{1}{2} \|\boldsymbol{y} - \mathscr{A}(\boldsymbol{X})\|_2^2 .\end{aligned}$$

We use alternating minimization method to solve the above optimization problem.

1. Use the data in the data file including $\boldsymbol{Z}_1^0, \boldsymbol{Z}_2^0$ to find $\boldsymbol{X}^1$. [3]
2. Use the obtained $\boldsymbol{X}^1$ to find $\boldsymbol{Z}_1^1$ and $\boldsymbol{Z}_2^1$. [3]
3. Implement the alternating minimization method. Run it until it converges. Draw your recovery. [4]

Part IV.

Coursework 3

8. Real Sparse Problems

8.1. Proximal Operator (Continued)

1. Use the data in the data file to compute the solutions of the following optimization problems.

   a)

   $$\min_{x} \frac{1}{2} \|x\|_2^2 + \frac{1}{2\gamma} \|x - z\|_2^2 .$$

   [1]

   b)

   $$\min_{x} \|x\|_2 + \frac{1}{2\gamma} \|x - z\|_2^2 .$$

   [2]

8.2. MRI CS Recovery: ADMM

Let X be an unknown MRI image. We consider the following MRI compressed sensing recovery formulation: [1]

$$\begin{aligned}\hat{X} = \arg\min_{X}\ & \frac{1}{2}\left\|y - \mathcal{P}_{\Omega} \circ \mathcal{S} \circ \mathcal{F}_{2\mathrm{D}}(X)\right\|_2^2 \\ & + \lambda_1 \left\|\mathcal{W}(X)\right\|_1 + \lambda_2 \left\|\mathcal{D}(X)\right\|_1 + \lambda_3 \mathrm{TV}(X),\end{aligned}$$

1: The formulation uses the combination of wavelet transform, DCT, and total variation. It has been observed in the literature that wavelet transform captures point-like features, DCT enhances homogeneous texture components, and total variations sharpens the edges.

where $\mathcal{P}_{\Omega}$ is the under-sampling operator, $\mathcal{S}$ is the shift operator, $\mathcal{F}_{2\mathrm{D}}$ is the two dimensional FFT transform, $\mathcal{W}$ is the two-dimensional discrete (Daubechies) wavelet transform (Julia package Wavelets), $\mathcal{D}$ is the two-dimensional discrete cosine transform (Julia package FFTW), and TV is the total variation defined as

$$\begin{aligned}\mathrm{TV}(X) &:= \sum_{m,n}\left((\nabla_1 x_{m,n})^2 + (\nabla_2 x_{m,n})^2\right)^{1/2} \\ &:= \sum_{m,n}\left((x_{m+1,n} - x_{m,n})^2 + (x_{m,n+1} - x_{m,n})^2\right)^{1/2}.\end{aligned}$$

In order to solve this problem using ADMM, we formulate the following constrained convex optimization problem:

$$\begin{aligned}\min_{X_{1:4}}\ & \frac{1}{2}\left\|y - \mathcal{A}(X_1)\right\|_2^2 \\ & + \lambda_1 \left\|\mathcal{W}(X_2)\right\|_1 + \lambda_2 \left\|\mathcal{D}(X_3)\right\|_1 + \lambda_3 \mathrm{TV}(X_1; X_4, X_5) \\ \text{s.t. } & X_2 = X_1,\ X_3 = X_1, \\ & X_4 = (X_1)_{2:m,1:n-1} - (X_1)_{1:m-1,1:n-1}, \\ & X_5 = (X_1)_{1:m-1,2:n} - (X_1)_{1:m-1,1:n-1},\end{aligned}$$

where we have introduced the linear operator $\mathcal{A}$ to replace $\mathcal{P}_{\Omega} \circ \mathcal{S} \circ \mathcal{F}_{2\mathrm{D}}$, and the notation $\mathrm{TV}(X_1; X_4, X_5)$ means the total variation of X_1 calculated from only using X_4 and X_5.

The corresponding augmented Lagrangian is then given by

$$\begin{aligned}L_{\rho} =& \frac{1}{2}\left\|y - \mathcal{A}(X_1)\right\|_2^2 + \lambda_1 \left\|\mathcal{W}(X_2)\right\|_1 + \lambda_2 \left\|\mathcal{D}(X_3)\right\|_1 \\ & + \lambda_3 \mathrm{TV}(X_1; X_4, X_5) \\ & + \langle U_2, X_2 - X_1\rangle + \frac{\rho}{2}\left\|X_2 - X_1\right\|_F^2 \\ & + \langle U_3, X_3 - X_1\rangle + \frac{\rho}{2}\left\|X_3 - X_1\right\|_F^2 \\ & + \left\langle U_4, X_4 - (X_1)_{2:m,1:n-1} + (X_1)_{1:m-1,1:n-1}\right\rangle \\ & + \frac{\rho}{2}\left\|X_4 - (X_1)_{2:m,1:n-1} + (X_1)_{1:m-1,1:n-1}\right\|_F^2 \\ & + \left\langle U_5, X_5 - (X_1)_{1:m-1,2:n} + (X_1)_{1:m-1,1:n-1}\right\rangle \\ & + \frac{\rho}{2}\left\|X_5 - (X_1)_{1:m-1,2:n} + (X_1)_{1:m-1,1:n-1}\right\|_F^2\end{aligned} \tag{8.1}$$

The data file provide the initial values for all variables and the values of the key parameters.

1. Write a function to update $\boldsymbol{X}_1$ when all other variables are fixed. Save the value of $\boldsymbol{X}_1^1$ into the data file. [5]
2. Write a function to update $\boldsymbol{X}_{2:5}$ when all other variables are fixed. Save the values of $\boldsymbol{X}_{2:5}^1$ into the data file. [12]
3. Write a function to update $\boldsymbol{U}_{2:5}$ when all other variables are fixed. Save the values of $\boldsymbol{U}_{2:5}^1$ into the data file. [4]
4. Now write a function to implement the ADMM algorithm including the stopping criteria mentioned in the lecture notes. Here we use the error tolerance constants $\epsilon_{\text{abs}} = \epsilon_{\text{rel}} = 10^{-4}$. The maximum number of iteration is 200. Save your final result $\hat{\boldsymbol{X}}$ into the data file. [5]
   Note that the hyperparameters $\lambda_{1:3}$ and ρ affect the critical point and convergence rate. In this particular part, you are allowed to change the hyperparameters in order to get good numerical results. Draw your final result $\hat{\boldsymbol{X}}$ as an image in the Jupyter notebook. [1]

8.3. Blind Deconvolution: Convex Relaxation

In this section, we observe the output of the cyclic convolution

$$\boldsymbol{y} = \boldsymbol{x} \circledast_n \boldsymbol{h}$$

and try to find the signal $\boldsymbol{x}$ and the channel $\boldsymbol{h}$ using convex optimization. The assumption is that $\boldsymbol{x}$ has small total variation

$$\mathrm{TV}(\boldsymbol{x}) = \sum_m |x_{m+1} - x_m| ,$$

and $\boldsymbol{h}$ is sparse and hence small l_1-norm.

We formulate the following convex optimization problem:

$$\begin{aligned}\min_{\boldsymbol{X}} \ & \frac{1}{2} \|\boldsymbol{y} - \mathcal{A}(\boldsymbol{X}_1)\|_2^2 \\ & + \lambda_1 \|\boldsymbol{X}_2\|_* + \lambda_2 \mathrm{TV}_{\mathrm{col}}(\boldsymbol{X}_1; \boldsymbol{X}_3) + \lambda_3 \|\boldsymbol{X}_4\|_{2,1} \\ \text{s.t. } & \boldsymbol{X}_2 = \boldsymbol{X}_1, \ \boldsymbol{X}_3 = (\boldsymbol{X}_1)_{2:m,:} - (\boldsymbol{X}_1)_{1:m-1,:}, \ \boldsymbol{X}_4 = \boldsymbol{X}_1,\end{aligned}$$

where $\|\cdot\|_*$ denotes the nuclear norm, $\mathrm{TV}_{\mathrm{col}}$ represents row-wise total variation

$$\mathrm{TV}_{\mathrm{col}}(\boldsymbol{X}_1; \boldsymbol{X}_3) := \sum_m \left\|(\boldsymbol{X}_1)_{m+1,:} - (\boldsymbol{X}_1)_{m,:}\right\|_2 = \left\|\boldsymbol{X}_3^{\mathrm{T}}\right\|_{2,1} ,$$

and $\|\cdot\|_{2,1}$ stands for the $l_{2,1}$-norm

$$\|\boldsymbol{X}\|_{2,1} = \sum_n \|\boldsymbol{X}_{:,n}\|_2 .$$

We shall solve the above problem using ADMM.

1. Write the explicit form of the augmented Lagrangian (similar to (8.1)). [1]
2. Write a function to update $\boldsymbol{X}_1$ when all other variables are fixed. Save the value of $\boldsymbol{X}_1^1$ into the data file. [4]
3. Write a function to update $\boldsymbol{X}_{2:4}$ when all other variables are fixed. Save the values of $\boldsymbol{X}_{2:4}^1$ into the data file. [9]
4. Write a function to update $\boldsymbol{U}_{2:4}$ when all other variables are fixed. Save the values of $\boldsymbol{U}_{2:4}^1$ into the data file. [3]
5. Now write a function to implement the ADMM algorithm including the stopping criteria mentioned in the lecture notes. Here we use the error tolerance constants $\epsilon_{\mathrm{abs}} = \epsilon_{\mathrm{rel}} = 10^{-4}$. Save your final result $\hat{\boldsymbol{X}}$ into the data file. [4]
   Note that the hyperparameters affect the critical point and convergence rate. In this particular part, you are allowed to change the hyperparameters in order to get good numerical results.

8.4. Blind Deconvolution

Similar to the last section, we observe the output of the cyclic convolution

$$\boldsymbol{y} = \boldsymbol{x} \circledast_n \boldsymbol{h}$$

and try to find the signal $\boldsymbol{x}$ and the channel $\boldsymbol{h}$. The assumption is that $\boldsymbol{x}$ has small total variation

$$\mathrm{TV}(\boldsymbol{x}) = \sum_m |x_{m+1} - x_m| \,,$$

and $\boldsymbol{h}$ is sparse and hence small l_1-norm. We also assume that $\|\boldsymbol{x}\|_2 = 1$.

We formulate the following optimization problem:

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{h}} \ & \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{x} \circledast \boldsymbol{h}\|_2^2 + \delta\left(\|\boldsymbol{x}\|_2 = 1\right) \\ & + \lambda_1 \|\mathrm{TV}\left(\boldsymbol{x}\right)\|_0 + \lambda_2 \|\boldsymbol{h}\|_0 \,.\end{aligned}$$

To proceed, we relax it into

$$\begin{aligned}\min_{\boldsymbol{x}_{1:3},\boldsymbol{h}_{1:2}} \ & \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{x}_1 \circledast \boldsymbol{h}_1\|_2^2 + \delta\left(\|\boldsymbol{x}_2\|_2 = 1\right) \\ & + \lambda_1 \|\boldsymbol{x}_3\|_0 + \lambda_2 \|\boldsymbol{h}_2\|_0 \\ & + \frac{\alpha}{2} \|\boldsymbol{x}_2 - \boldsymbol{x}_1\|_2^2 \\ & + \frac{\alpha}{2} \|\boldsymbol{x}_3 - ((\boldsymbol{x}_1)_{2:m} - (\boldsymbol{x}_1)_{1:m-1})\|_2^2 \\ & + \frac{\alpha}{2} \|\boldsymbol{h}_2 - \boldsymbol{h}_1\|_2^2 \,,\end{aligned}$$

where $\alpha > 0$.

We solve the above optimization problem using alternating minimization.

1. With fixed $\boldsymbol{h}_{1:2}$, we update $\boldsymbol{x}_{1:3}$ using proximal gradient method (with only one iteration).
   a) Find the Hessian matrix of
   $$\frac{\alpha}{2} \|\boldsymbol{x}_2 - \boldsymbol{x}_1\|_2^2 + \frac{\alpha}{2} \|\boldsymbol{x}_3 - ((\boldsymbol{x}_1)_{2:m} - (\boldsymbol{x}_1)_{1:m-1})\|_2^2 \,.$$
   [3]
   b) Find the upper bound (τ_{ub}) of the step size in the proximal gradient step to update $\boldsymbol{x}_{1:3}$. [1]
   c) Choose the step size as $0.8\tau_{\mathrm{ub}}$ and update $\boldsymbol{x}_{1:3}$ using one iteration of the proximal gradient method. Save your result into the data file. [2+2+2=6]
   d) Evaluate the subgradient with respect to $\boldsymbol{x}_{1:3}$ [3]
2. With fixed $\boldsymbol{x}_{1:3}$, we update $\boldsymbol{h}_{1:2}$ using proximal gradient method (with only one iteration).

a) Find the Hessian matrix of

$$\frac{\alpha}{2} \|\boldsymbol{h}_2 - \boldsymbol{h}_1\|_2^2$$

[1]

b) Find the upper bound (τ_{ub}) of the step size in the proximal gradient step to update $\boldsymbol{h}_{1:2}$. [1]

c) Choose the step size as $0.8\tau_{\text{ub}}$ and update $\boldsymbol{h}_{1:2}$ using one iteration of the proximal gradient method. Save your result into the data file. [2+2=4]

d) Evaluate the subgradient with respect to $\boldsymbol{h}_{1:2}$ [2]

3. Evaluate the subgradient with respect to $\boldsymbol{x}_{1:3}$ using the updated $\boldsymbol{x}_{1:3}$ and $\boldsymbol{h}_{1:2}$. [2]
4. Complete the alternating minimization method where each iteration involves one proximal gradient step to update $\boldsymbol{x}_{1:3}$ and that to update $\boldsymbol{h}_{1:2}$.
   Include that the number of alternating minimization is at most 500 into your stopping criteria.
   Save your results into the data file.
   Note that the hyperparameters affect the critical point and convergence rate. In this particular part, you are allowed to change the hyperparameters in order to get good numerical results. [4]

Part V.

Coursework 4

9. Learn and Tell Coursework Guideline

The purpose of this coursework is to encourage students to

- Explore the literature, find interesting or useful stuff to study and present;
- Learn from each other and broaden horizons;
- Practice skills of project management, decision-making, team-playing, leadership, and presentation.

9.1. Coursework Format

Students will work in groups and present a technical topic that they have learned. The topic must be relevant to the module but should go beyond the taught materials. It can be on application or/and theory. Coursework will be marked by GTAs, the instructor, and peer students.

The **milestones** for this coursework are as follows.

1. We will provide detailed information on coursework 4 presentations according to Table 1.1.
2. Students need to form their groups, decide a topic, and choose a time slot for their presentation. The relevant information should be input in the relevant excel file by the due date shown in Table 1.1.
3. Groups submit their slides to the Teams folder by the due time (Table 1.1). Make sure that the key references are listed in your slides. After the due time, the Teams folder will be read-only.
4. Presentation:
    - Each group will have 15 minutes for presentation and 5 minutes for questions.
    - The group is required to attend all the presentations in the session where they present.
    - All the groups in a session are required to arrive 10 minutes before the session starts for the setup.

9.1.1. Presentation Sessions

The presentation sessions will be finalised later. The following list is tentative.

- 4 December, 2023, Room 611
    - 9:30-11:10
    - 11:30-13:10
- 5 December, 2023, Room 909B
    - 16:00-17:40

9.2. Marking Scheme

$$M_i = (0.4M_{g,p} + 0.6M_{g,a}) * 0.92 + M_{i,p} * 0.08$$

where

- $M_{g,p}$ is the mark for the group g from peer marking. Each student is supposed to mark certain number of presentations given by other groups. The peer marks will be aggregated.
- $M_{g,a}$ is the mark for the group g from 'authoritative' marking. 'Authoritative' marks are given by GTAs and the instructor.
- $M_{i,p}$ is the mark for the individual student i, depending on the quality of the student's peer marks/comments for other groups. Note that every student is required to mark at least 7 other groups.

9.2.1. Marking Criteria for Presentations

- Technical topic and contents (50%): Timeliness, relevance, depth, and breadth. (Something interesting and useful)
- Presentation and delivery (35%): Clearness, conciseness, and easiness to follow. Time control. (Audience can learn something)
- Q&A (15%): Accuracy and conciseness. Time control.

9.3. Topic Selection

- Every group finds their own topic to present. The topic can be theory, methodology, or applications.
- The reading materials can be academic papers, book chapters, published articles, etc.
- Typical questions to be answered when reading a paper include
  - What is the problem under study? Why is the problem relevant/important?
  - What are the approaches/methods to address the problem?
  - What are the pros and cons?

9.3.1. Possible Topics

The following list of topics is only for your references. You can go beyond.

- Sparsity related problems
  - Blind deconvolution
  - Blind image deblurring
  - Blind source separation
  - Causality analysis
  - Channel estimation for massive MIMO
  - Collaborative filtering for recommender systems
  - Convolutional dictionary learning
  - Compressed sensing and physics informed deep Learning
  - Dictionary learning

- Graph neural networks
- Graph signal processing
- Group Lasso and network Lasso
- Landscape of neural networks
- MRI imaging via compressed sensing / deep learning
- Non-negative matrix factorization
- Pagerank in practice
- Prototypical learning (part-based prototypes)
- Resource allocation for wireless communications
- Sparse logical models: decision trees, decision lists, and decision sets
- Sparse PCA
- Super resolution
- SVM regression

▸ Optimization

- Conic programming
- Conjugate gradient method
- Minimax optimization
- Mirror descent
- Nesterov accelerated algorithms
- Newton type proximal gradient methods
- Optimizers for neural networks
- Proximal algorithms
- Semidefinite programming
- Stochastic optimization
- Trust region methods
- Zero-order optimization